Afterburners

Erin Sheldon (NYU)

Def: A secondary burner fitted to increase thrust

Data retrieval and archiving

- Afterburners need to extract images and corresponding object lists.
- Most afterburners will work simultaneously on all observations of an object or region of the sky.
- Examples: co-add, shapelet pipeline

Example: Shapelets

- For a region of sky:
  - Extract all overlapping images.
  - Extract object lists and matching information
  - PSF information for each image. Either pre-computed interpolation at each object or the PCA information with which to calculate it.
  - Measure properties from all images containing a unique object. Stuff measurements into DB with unique id.

Issues

- Image extraction
- Deblending
- Database transactions

Image Extraction

- Will it be simple and efficient to extract all images for a given region of sky?
- Need images grouped by sky region rather than, say, observation date. Re-structure archive to optimise?
- Or will we be working with cutouts (atlas images) stored in database itself? Cutouts in original image will not exist for objects detected only in the co-add.

Deblending

- Better to re-deblend for each afterburner?
- Co-add will have its own deblend.
- Deblending choices in original images may differ when co-add information is available
- Again, associated cutouts would have to be created based on the co-add

Best route

- These points favor going back to original images if possible.
- Be opportunistic? Perform co-add followed immediately afterburners if possible to re-use retrieved images.
- What about re-processing?

Database

- Database ingestion: for flexibility, we would rather not write a new stuffer for each afterburner in C.
- Need a generic routine to ingest either from memory or a CSV/FITS file.
- Format unknown in advance: easiest in Python, or some interpreted language.
- Binary bulk input most efficient, with caveat that table still must be re-indexed.